UNITED STATES PATENT AND TRADEMARK OFFICE

TD

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/738,542 | 12/17/2003 | Omar Dewan | IDF 2565 (4000-15800) | 8268 |

28003 7590 12/27/2006
SPRINT
6391 SPRINT PARKWAY
KSOPHT0101-Z2100
OVERLAND PARK, KS 66251-2100

| EXAMINER | |
|---|---|
| LU, KUEN S | |
| **ART UNIT** | **PAPER NUMBER** |
| 2167 | |

| SHORTENED STATUTORY PERIOD OF RESPONSE | MAIL DATE | DELIVERY MODE |
|---|---|---|
| 3 MONTHS | 12/27/2006 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

| *Office Action Summary* | Application No. | Applicant(s) |
|---|---|---|
| | 10/738,542 | DEWAN ET AL. |
| | **Examiner** | **Art Unit** |
| | Kuen S. Lu | 2167 |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *24 October 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 September 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

## Detailed Action

### *Response to Amendment*

**1.** This is responsive to Applicant's Amendment filed October 24, 2006. Applicant's amendments made to Specification and claims 1, 4, 5, 20, 22, 27 and replacement of drawings are acknowledged. Objections to claim 5, Specification and drawings are hereby withdrawn. Withdraw of claim rejections under 35 U.S.C. § 112 and 35 U.S.C. § 101 is also necessitated by the amendments made to each of independent claims 1, 20 and 27.

**2.** As to Applicant's Arguments/Remarks filed October 24, 2006, please see Examiner's response in "***Response to Arguments***", following this Office Action for Final Rejection shown next. Please note claims 1-32 are pending.

### *Claim Rejections - 35 USC § 103*

**3.** The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

**3.1.** Claims 1-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Christof Dallermassl: Aspects of Integration of Heterogeneous Server Systems in

Intranets – the JAVA Approach, Graz University of Technology, Graz, November 1999 (hereafter "Dallermassl") in view of Goodman et al. (U.S. Patent 7,020,697, hereafter "Goodman").

As per claim 1, Dallermassl teaches "A naming service for locating a service in an enterprise" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"a first module operable to maintain a location of an interface, the interface having a reference to a service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service).

Dallermassl does not explicitly teach "a second module operable to provide the location of the interface to an application in response to receiving a request from the application for the location of the service", although Dallermassl teaches JNDI defining and supporting hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services at Fig. 3.3 and Pages 26-27, Para. 3.4.2.

However, Goodman teaches "a second module operable to provide the location of the interface to an application in response to receiving a request from the application for the location of the service" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach a second service providing node when a service node is down, and the service includes directory

service).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Goodman with Dallermassl because location transparency in service providing would have fulfilled the requirements of durability, consistency and isolation for transactions that are a vital point in Dallermassl's middleware systems.

Goodman further teaches "wherein the naming service provides service location transparency such that the location of the service can be changed without effecting the behavior of the application" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach another service providing node when a service node is down, and the service includes directory service).

As per claim 20, Dallermassl teaches "An enterprise naming service for applications to locate services" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"a binding module to associate a first service with a location of an interface maintaining a reference to the first service, the binding module further operable to associate a second service with a location of <u>an interface maintaining a reference to</u> the second service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service).

Dallermassl does not explicitly teach "a look-up module operable to provide the location of the interface of the first service to a first application in response to a request by the first application for the first service, the look-up module further operable to provide the location of the interface of the second service in response to a request by a second application for the second service", although Dallermassl teaches JNDI defining and supporting hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services at Fig. 3.3 and Pages 26-27, Para. 3.4.2.

However, Goodman teaches "a look-up module operable to provide the location of the interface of the first service to a first application in response to a request by the first application for the first service, the look-up module further operable to provide the location of the interface of the second service in response to a request by a second application for the second service" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach a second service providing node when a service node is down, and the service includes directory service).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Goodman with Dallermassl because location transparency in service providing would have fulfilled the requirements of durability, consistency and isolation for transactions that are a vital point in Dallermassl's middleware systems.

Goodman further teaches "wherein the enterprise naming service provides service

location transparency such that the location of a service can be changed without effecting the behavior of an application" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach another service providing node when a service node is down, and the service includes directory service).

As per claim 27, Dallermassl teaches "A method for locating a service in an enterprise" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service), comprising:

"associating a service with a location with an interface maintaining a reference to a service" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service).

Dallermassl does not explicitly teach "requesting, by an application desiring to employ the service, the location of the service", although Dallermassl teaches JNDI defining and supporting hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services at Fig. 3.3 and Pages 26-27, Para. 3.4.2.

However, Goodman teaches "requesting, by an application desiring to employ the service, the location of the service" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach a second service providing node when a service node is down, and the service includes directory

service).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Goodman with Dallermassl because location transparency in service providing would have fulfilled the requirements of durability, consistency and isolation for transactions that are a vital point in Dallermassl's middleware systems.

Dallermassl further teaches "returning the location of the interface to the application" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system to JNDI and enabled to connect all directory services via JNDI).

Goodman further teaches " "wherein the method provides service location transparency such that the location of a service can be changed without effecting the behavior of an application" (See col. 94, line 64 – col. 95, line 4 and col. 107, lines 53-62 where service call is location transparent and the call can reach another service providing node when a service node is down, and the service includes directory service).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Goodman with Dallermassl because location transparency in service providing would have fulfilled the requirements of durability, consistency and isolation for transactions that are a vital point in Dallermassl's middleware systems.

As per claim 2, Dallermassl further teaches "wherein application is operable, using the location of the interface, to the service using the interface" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI is the interface, and a Dino is implemented as an external embedded system being enabled applications to connect all directory services).

As per claim 3, Dallermassl further teaches "the service is further defined as a service object" (See Fig. 3.1 and Page 18, last Paragraph where application requests an operation performed by distributed object and result returned in a CORBA architecture).

As per claim 4, Dallermassl further teaches "the service object is further defined as a JAVA service object and wherein the interface is further defined as a JAVA directory and naming interface" (See Fig. 33, and Pages 20 and 26 where JAVA objects at remote hosts are invoked by JAVA application and JDNI is an naming interface).

As per claim 5, Dallermassl further teaches "the service object is selected from a group of service objects including an enterprise JAVA Bean, a queue, and a queue manger" (See Page 45, Para. 4.3.3. and Page 54, Para. 5.2.1 where application offers Enterprise JAVA Bean environment and Dino system was message based having events queued in global message queue and listened by interested components).

As per claim 6, Dallermassl further teaches "the first module id further operable to maintain a second location associates with a second service, the second module further

operable to provide the second location to a second application in response to receiving a request for the second service from the second application, the second application using the second location to use the service" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access RMI registry naming service and RMI registry, and RMI enables JAVA application to invoke method of JAVA objects on remote hosts).

As per claim 7, Dallermassl further teaches "the second service is a service object and wherein the second location is useful by the application to directly invoke the services" (See Fig. 4.3, and Pages 20 and 43 where RMI enables JAVA application to invoke method of JAVA objects on remote hosts).

As per claim 8, Dallermassl further teaches "second location is selected from a group of locations including an address and reference location" (See Fig. 4.3, and Pages 20 and 43 where RMI enables JAVA applications to invoke methods of JAVA objects on remote hosts).

As per claim 9, Dallermassl further teaches "first module further maintains an identifier corresponding to the service and associates the identifier with the location of the interface" (See Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space).

As per claim 10, Dallermassl further teaches "the identifier is a service type of the service" (See Page 27, first paragraph where objects stored in directory are of type).

As per claim 11, Dallermassl further teaches "the identifier includes is a name and a service type associated with the service" (See Pages 27, first paragraph and 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space and system provides names for the objects).

As per claim 12, Dallermassl further teaches "the interface is a server and the application is a client of the server, the client using the server to provide the service to the client" (See Figs. 3.1-3.2 and Pages 17-18, Para. 3.1.1 and Pages 20-21, Para. 3.1.2 where applications are running at clients and server provides services via interface in the CORBA architecture).

As per claim 13, Dallermassl further teaches "the second module returns meta-information to the application, the meta-information including a reference useful by the application for employing the service" (See Page 20, Para. 3.1.2 where JAVA application invokes method of remote JAVA objects and receives reference which is a metadata).

As per claim 14, Dallermassl further teaches "the service is a server maintaining a plurality of classes and a plurality of objects, at least one of the objects useful by the application" (See Page 20, Para. 3.1.2 where JAVA application invokes method of remote JAVA objects and Page 3.3, Para. 3.3 where JDBC is a JAVA API consists of a set of classes).

As per claim 15, Dallermassl further teaches "first module stores the location of the interface in a datastore" (See Page 3.3, Para. 3.3 where JDBC is a JAVA API consists of a

set of classes and interfaces for accessing a database management system).

As per claim 16, Dallermassl further teaches "the datastore is further defined as a lightweight directory access protocol based datastore" (See Pages 26-27, Para. 3.4.2 where Dino system uses JNDI to connect directory services, including LDAP).

As per claim 17, Dallermassl further teaches "including a third module operable to store a service status information related to the service, the third module operable to search and return the service status information related to the service in response to a request" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service).

As per claim 18, Dallermassl further teaches "a hypertext markup language interface is employed to communicate with the third module" (See Page 20, Para. 3.1.2 where RMI use HTTP for network communications).

As per claim 19, Dallermassl further teaches "the third module is defined as a name service browser" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service).

As per claim 21, Dallermassl further teaches "the first service is a service object and the second service is a Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to

access RMI registry naming service and RMI registry, and RMI enables JAVA application to invoke method of JAVA objects on remote hosts).

As per claim 22, Dallermassl further teaches "the interface is a JAVA Naming and Directory Interface and wherein the first service is a JAVA service object" (See Fig. 33, and Pages 20 and 26 where JAVA objects at remote hosts are invoked by JAVA application and JDNI is an naming interface).

As per claim 23, Dallermassl further teaches "a name service browser module operable to maintain a service status information related to one of the first and second services, the name service browser operable to search and return the service status information of one of the first and second services in response to a request" (See Fig. 4.3, and Pages 20 and 43 where Voyager ORB offers API to access COS naming service and at Fig. 4.2 and Pages 39-40, Para. 4.2.2 where in CORBA architecture Web browser browsing services responding client request).

As per claim 24, Dallermassl further teaches "the binding module is further operable to maintain a version identifier associated with at least one of the first and second are maintained" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled).

As per claim 25, Dallermassl further teaches "the look-up module is further operable to return the location associated to a first version of the first service" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled, similar to a CVS, and supporting checking in and out, locking, retrieving, tagging and merging of specific version).

As per claim 26, Dallermassl further teaches "the look-up module is further operable to return the location associated to a first version of the second service" (See Page 14, Para. 2.9 where objects stored in Dino system are version controlled and supporting checking in and out, locking, retrieving, tagging and merging of specific version).

As per claim 28, Dallermassl further teaches the following:

"using the location to communication between the application and the interface" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service);

"requesting, by the application, the service from the interface" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services); and

"using the service by the application" (See Fig. 3.3 and Pages 26-27, Para. 3.4.2 where a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system to JNDI and enabled to connect all directory services via JNDI).

As per claim 29, Dallermassl further teaches “the application uses a service identifier to request the location of the service” (See Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space).

As per claim 30, Dallermassl further teaches “the service is defined as a service object and wherein the interface is further defined as a naming and directory interface” (See Fig. 33, and Pages 20 and 26 where JAVA objects at remote hosts are invoked by JAVA application and JDNI is an naming interface).

As per claim 31, Dallermassl further teaches “the interface is defined as a JAVA Naming Directory Interface and the service is an Enterprise JAVA Bean, the method further comprising associating an identifier, a version and a second location with a Common Object Request Broker Architecture object” (See Page 45, Para. 4.3.3. and Page 54, Para. 5.2.1 where application offers Enterprise JAVA Bean environment and Dino system was message based having events queued in global message queue and listened by interested components and at Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service).

As per claim 32, Dallermassl further teaches the following:

"requesting the second location of the Common Object Request Broker Architecture object using the identifier and version of the Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Page 14, Para. 2.9 where objects stored in Dino system are version controlled);

"returning the second location of the Common Object Request Broker Architecture object based on the identifier and version of the Common Object Request Broker Architecture object" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service, Page 75, Para. 5.4.4 where applications access the Dino system by use of its API and its views, and internal IDs are used to address objects in the Dino space, and Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services);

"connecting to the Common Object Request Broker Architecture object using the second location" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services); and

"employing the Common Object Request Broker Architecture object at the second location" (See Fig. 4.3 and Page 43 where Voyager ORB offers API allowing objects to communicate with CORBA naming service and at Fig. 3.3 and Pages 26-27, Para. 3.4.2 where JNDI defines and supports hierarchical structures of objects by using naming and directory services and having objects stored in directory, and a Dino, Distributed Interactive Network Objects, is implemented as an external embedded system being enabled to connect all directory services).

***Response to Arguments***

**4.** Applicant's arguments with respect to currently amended claims 1, 20 and 27, Examiner respectfully submits that an introduction of Goodman reference providing the teaching to set forth grounds for claim rejections under 35 U.S.C. 103(a). During a telephone communication, Examiner suggested a proposed Examiner's amendment might be feasible for submitting the application for potential allowance review and approval. After a further consideration, Examiner regretted the suggestion is not currently feasible.

**5.** The prior art made of record

U. Christof Dallermassl: Aspects of Integration of Heterogeneous Server Systems in Intranets – the JAVA Approach, Graz University of Technology, Graz, November 1999.

E. U.S. Patent No. 7,020,697

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

V. Barlen et al.: Implementation and Practical Use of LDAP on the IBM @server iSeries Server, April 2002, IBM

A. U.S. Patent Application 2003/0018701

B. U.S. Patent No. 7,036,127

C. U.S. Patent Application 2005/0015401

D. U.S. Patent No. 5,987,471

F. U.S. Patent No. 6,510,450

***Conclusion***

**6.** Applicant's amendment necessitated the new grounds of rejection presented in this Office Action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1 .136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

***Contact Information***

**7.** Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Kuen S. Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 703-305-39000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 703-305-3900 (toll-free).

Kuen S. Lu
Patent Examiner, Art Unit 2167

December 22, 2006

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100